GUMNAM
"Name : anonymous"

Abhilasha Bhargav (bhargav@purdue.edu) Sarvjeet Singh(sarvjeet@purdue.edu)

CS 555 –PROJECT REPORT

Aim:

Our aim was to implement a two way anonymous communication protocol. In this protocol, unlike the other protocols used for anonymity, identity of both the sender and receiver is to be protected. As, a communication cannot be initiated without the knowledge of the second party involved in it, we use a concept of public handle to identify the party that we are trying to reach. We are using a semi trusted third party for this protocol and have tried to find a minimal trust model for the communication.

1 Motivation:

1.1) Medical Surveys:

Surveys of this nature require data to stay anonymous by stripping identifiers. Although to be useful for data analysis certain identifiers (as defined by the regulation) are critical. Currently this requires trusted third party data collectors. In addition to stripping of basic identifiers providing anonymity at an earlier stage i.e. during the submission of their data may be useful. Our current implementation is 2 way anonymous i.e. both parties do not know each other. But in the above example only the people submitting the survey forms are anonymous and not the company collecting it. At the same time, for example, a medical researcher using the World Wide Web may expect his particular focus to remain private, and could conduct surveys where his identity is hidden. This could be implemented with our two way anonymous communication.

1.2) Privacy on the internet:

Health information on the internet requires anonymity of the web user, which is currently dependent on the policies of the website.
Individuals may wish to protect their privacy too. The identities of the participants in an e-mail conversation should be known to each other only with respect to the "handle" (which could be the topic of discussion/ sharing a resource or information etc.)
A very popular solution is Onion routing. It has been prototyped on Sun Solaris 2.4. The prototype includes proxies for private Web browsing, remote login, e-mail, and file transfer protocols, and also anonymizing versions of those proxies that remove identifying information from the data stream.(JYA)
How our protocol is different from Onion Routing is that both parties do not know each other. In onion routing applications and users may identify themselves to each other. But the use of a public network should not automatically reveal to others the identities of communicating parties.

1.3) Information Sharing between companies

The efficiencies of the public Internet are strong motivation for companies to use it instead of private intranets. However, these companies may want to protect their interests. The existence of inter-company collaboration may be confidential. Current solutions exist which are mainly related to data mining where the result of queries is such that no more information than that requested can be inferred from the result of the query. This is different from the problem we are approaching because here we do not know which companies are sharing information and hence the nature of the information can be very different.

2 Simplified Description the Model:

The model consists of two main entities.

- Post Office
- People

The post office has a number of mailboxes. The first mailbox is a public message board equivalent a real world post office. In the real world any person can write any message and pin it up on a message board. The identity of the person is not given in the message and so the anonymity could be achieved. But if the post master of the post office is malicious, he would see what message the person is posting and identify the message with the person thus breaking the anonymity.

Now if the people decide that since writing the message individually does not work they will take a big sheet of paper and pass it around with each person writing his message in any one line of the paper. Any one of them could post this message to the message board. This time even if the post master sees the person posting message he cannot identify any person in the group with the message.

What if some of the people in the group are malicious, in the sense want to find out the message of the other person as the paper is being passed around. This is prevented by blinding all the lines in the paper as they pass it around to each other. So there is a possibility of overwriting somebody else's message in which case the paper has to be passed around to each person in the group again. Therefore the size of the paper to be chosen should be big enough to make sure that the probability of two people writing in the same place is less.

The above described what we call as the ***anonymous write***. Detailed description is given in section

Since we want a two way communication there should be a way such that a person who wants to reply to the anonymous post can do so successfully. For this the message posted will also contain a secret key and a mailbox in the post office where the message for this anonymous person should be posted. So the replier does again an

"anonymous write" to the specified mailbox with his reply locked in the key given such that only the person who posted the original message can open the message. Many people can request their messages to be sent to the same mailbox which is a factor when a reply is being sent to this mailbox. Thus we achieve a two way anonymous communication.

3 Trust Model:

One of our aims was to achieve a minimum trust model for the anonymous

3.1) Post office:

The Postoffice is trusted to follow the protocol exactly. It is not trusted not to try to link the identity of a person with a given message. Hence we call the Postoffice "semi-trusted".

3.2) People:

The people involved in writing messages (anonymously) are trusted to follow the protocol. A person within this group may at anytime during the protocol may try to attack the model to find the connection between the identity and message of any of his peers. Hence in this we the people involved in the protocol are also semi trusted.

Please note that we aim to make the level of trust for the whole group the minimum. This means that we need to trust (to follow the protocol) only a certain percentage of the group sending the message which would satisfy the security requirement. Further analysis is required to determine this threshold.

The external world (not involved in the protocol) is totally un trusted and attacks of any form can be launched by them.

4 Protocol Components

4.1) Postoffice:

{

1) Post Master:

The one who carries out the protocol and talks directly to the people who want to write in any of the mailboxes in the Post Office

2) Mail Boxes:

Mailboxes are of 2 kinds

Public: Mailboxes with unencrypted messages which can be read by anyone. This corresponds to a message board in a post office. In our implementation we have one such mailbox.

Private: Mailboxes with encrypted messages.

}

4.2) People:

People: **P** = {set of people having mailboxes in the Postoffice}
Requester: R = {set of people requesting for an information}
Provider: O = {set of people who are sending information to those who requested}

4.3) Level of anonymity:

This is a parameter defined by the person. This is directly proportional to the level of anonymity that they are requesting.

5 Anonymous Write:

The key to achieve anonymity is being able to write to the Server without being identified with the message. This is achieved by writing to the server in groups. The size of the group aids in the level of anonymity accomplished.

A simple example to illustrate anonymous write:

Say the level of anonymity = 3;

1. P1 informs the Postmaster that he wants to write to the message board.
2. Server checks the number of requests he has to write to the message board. It is currently 0. He increments this number and asks P1 to wait.
3. P2 informs the Postmaster that he wants to write to the message board.
4. Server checks the number of requests he has to write to the message board. It is currently 1. He increments this number and asks P2 to wait.
5. P3 informs the Postmaster that he wants to write to the message board.
6. Server checks the number of requests he has to write to the message board. It is currently 2. He increments this number and makes a ring of the there people P1, P2 and P3.
7. Currently we are using RSA encryption scheme. Postmaster gives P1 key K1 = Public key of P2 multiplied by Public Key of P3 = E_{P2*P3}. It gives P2 the public key E_{P3}. It also gives P1 a message block of size 6 (6 messages can be successfully written to the block.)
8. P1 decrypts the whole block with his private key encrypts his message with the key given by the postmaster and then puts this message in a random slot of the message block. He then passes it to P2.
9. P2 decrypts the whole block with his private key encrypts his message with the key given by the postmaster and then puts this message in a random slot of the message block. He then passes it to P3.
10. P3 decrypts the whole block with his private key does not encrypt his message and then puts this message in a random slot of the message block. He then passes it to P1.
11. P1 verifies if his message is still there in the message block. If it is then he simply passes this message block to P2 with the message saying Yes. If it is not then it does what it did in step 8 and gives the message No to P2.

12. If the message P2 got was Yes then P2 simply verifies his message and passes on the message block to P3 with an appropriate answer. If the message P2 got was No then he executes the step 9 of the protocol.
13. P3 does the same. If Yes was given all the way that means all the intended messages are successfully copied to the block and is ready to be given to the server. If there was any no the verification step is repeated again.
14. When the block is ready to be given to the Postmaster P3 sends this block to him and the Postmaster then writes to the message board.

The above was a simple example to illustrate how we hope to achieve anonymity while writing to the main post office. The message being written is either encrypted or plain text depending on function. For example like in above the message will be plaintext because it is simply a message to be posted on the message board. If on the other hand if it were to be a reply to such a request then the provider would encrypt the message with a key specified by the requester and would be written to the mailbox specified by the requester. We think that this part of the project; the *anonymous write*, is the key to the anonymity provided in both directions. The read is anonymous by the nature of the function.

6.1) Space Complexity:

The size of the block is chosen in such a manner such that the probability of the message of a previous person in the ring is not overwritten.

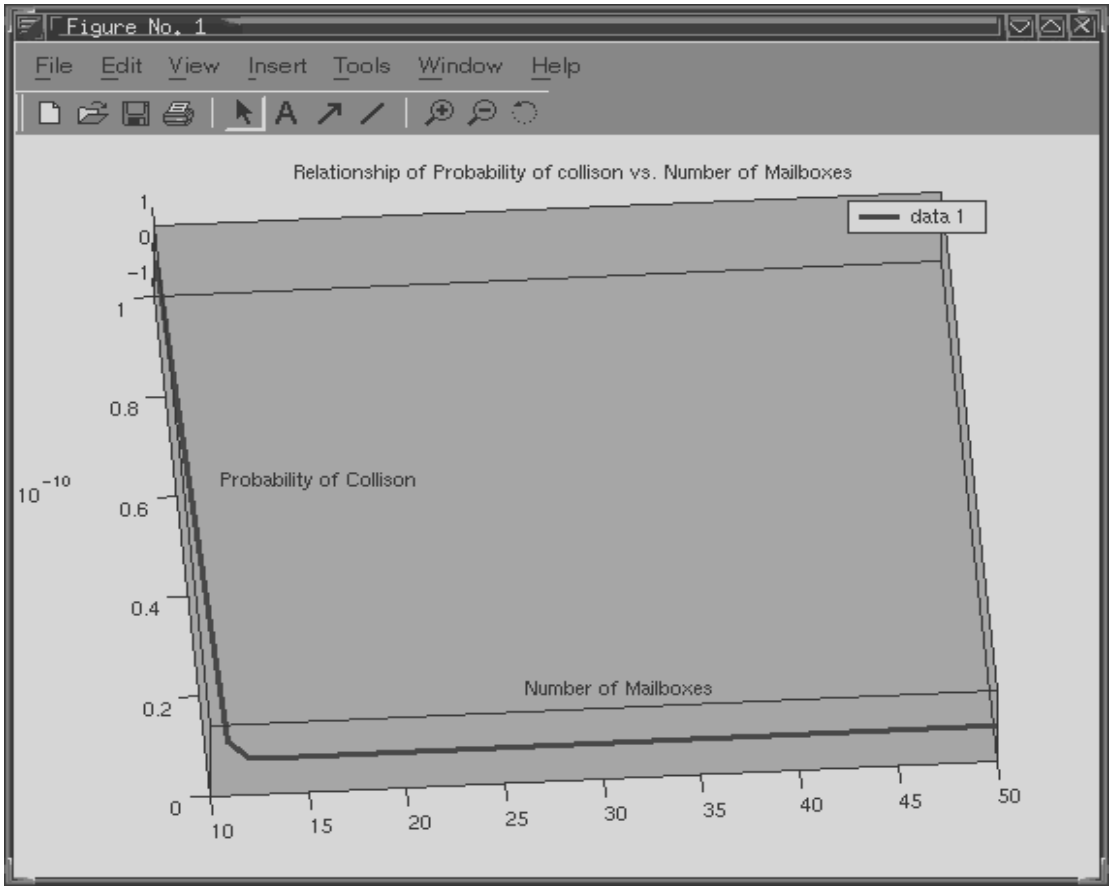
If

Number of slots in the block = m;
Number of persons writing = n;
Probability of not overwriting = $m*(m-1)*\ldots(m-n+1)/m^n$
$>= (m-n+1)^n/m^n$

If we want this probability to be greater than equal to 1/2 we get
$(1 - (n+1)/m)^n >= ½$
$m >= (n+1) * 2^{n/2}/(2^{n/2} -1)$

Like in the above example when n = 3 then from the above calculation we get n approximately equal to 19. This is because the relationship between n and m is exponential. We think that this is a big drawback in our project and future work is to replace this block with alternatives like shared memory etc.

As the figure below shows the relationship between the number of mailboxes and probability of collision is exponential in the current model.



6.2) Time Complexity:

Each step of the anonymous write consists of

1) Decryption of all slots of the block O(Em) if m is the number of slots in the block and O(E) is the time taken for the encryption.
2) 1 encryption if the message is required to be encrypted O(E).
3) Constant time to pass the message.

Thus overall the total run time of the write is O(Emn). Again we think that this time can do better if other alternatives are explored.

6.3 Fault Tolerance:

We want each person to follow the protocol as specified. No person in the ring can relate the pervious person in the ring with an entry in the block which generally looks random. If in a ring of n people n-1 people collude then when the messages are out in the clear (which is when it reaches the last person in the ring) then it can easily identify the message of the one person against whom the people are colluding.

The relationship between the number of people in the ring and the number of slots in the mailbox is almost exponential as seen above. This can also be advantageous: given a small subset of honest people in the group it will be still very hard for the dishonest people to relate the message with the people in the honest group. Thus this tries to achieve a minimal trust model where even the postmaster is not trusted totally.

6.4 Scalability:

Since most of the work is done by the persons the postmaster is left free to take care of other requests. The work is well distributed among each person in the ring formed and they do not have to do a lot of bookkeeping to execute the protocol.

Our current implementation:

We are using Remote Procedure Call (RPC) in C to implement the above system. In our application, we have implemented an anonymous chatting system in which the participants do not want to reveal their identity. Their identity is protected both from the chat server (Postmaster) and the chat clients (Persons). Each person posts an alias on the public notice board and other persons can either send a one way message to that person or initiate a two way conversation (by giving their alias in the first message).

Number of mailboxes = 2; One is the public notice board and one a private mailbox.
The current security policy (level of anonymity) is hard coded (= 2) which can be altered.
Postmaster is a RPC server providing remote methods like read and write.
Persons are both RPC server providing remote methods to implement the write protocol. The persons program also acts as RPC client for other person programs.

<u>Modules Descriptions:</u>

- Encryption Module: This handles the encryption and decryption of the messages using public-private key. We have currently implementing the RSA C functions from the files downloaded from openssl.

- Server: This module is the central semi-trusted (in the sense that it will at least follow the protocol) postmaster that we have mentioned above. It has the following sub modules:

    - Security policy: This manages the parameters which in turn will define the level of anonymity of the system as a whole. This security policy also incorporates security policies of the clients. For example, for an anonymous write, this policy will define the minimum number of writers that should be allowed to write simultaneously so that the writer anonymity is not compromised.

    - Read: This module responds to the read requests of the clients. This module returns the contents of mailbox requested by the client.

- Write: Similar to read, this responds to write requests by the clients. It applies the security policies and coordinates with a number of writers to implement the write protocol. For example, if client A wants to write to mailbox number 'p', and security policy requires the minimum number of writers to be 'n' then it will wait for n-1 more writers to come and then initiate the write protocol.

- Client: There could be any number of clients which will connect to the server and send/receive messages. These are the sub-modules:

  - Security policy: This handles the security parameters on client's side. For example, client may not want to write if the number of participants in the write protocol is less than 10. These parameters are communicated to the server's security policy when the client connects to the server.

  - Read: This module reads messages from mailboxes as well as the public notice board from the server.

  - Write: This is the main function. This provides the anonymous write facility for the client. For example, if the client wants to write to mailbox number 'p', this module will send the write request to server's write module. As mentioned above, the server's write module will wait and form a group of writer clients and send the clients the group information. The clients will now initiate the write protocol (in which they form a ring) and write back to the server.

Conclusion:

We proposed to implement a 2 way anonymous message exchange. To do that we read through some papers initially ranging from current implementations like that of onion routing, and some theoretical papers on anonymity. We have tried to achieve a minimal trust model. Although efficiency was also one of the goals, we have to work more to find better ways of doing anonymous message write.
There are some nice differences in the given implementation as compared to the current solutions. It is general and can be easily used for specific 2-way anonymous communication for some purpose (chatting, resource sharing etc.).
As future work we hope to find a good solution to decrease space requirement and increase efficiency of the current anonymous write to make the protocol feasible in real world. Studying more papers will be helpful. There are several links in the reference that we came across during the semester, some of which we studied better than the others but have listed here for future reference. The current implementation can be built on to achieve complete functionality. More attacks and risk analysis needs to be done for this model.

Reference Material:
http://citeseer.nj.nec.com/freedman02tarzan.html
A Peer-to-Peer Anonymizing Network Layer, M.J. Freedman, Thesis,MIT 2002

http://citeseer.nj.nec.com/clarke00freenet.html
Freenet: A Distributed Anonymous Information Storage and Retrieval System (2000)

www.cs.washington.edu/research/networking/websys/pubs/usits-submit.ps
Organization-Based Analysis of Web-Object Sharing and.. - Wolman, Voelker, al. (1999)

Group Principals and the Formalization of Anonymity - Syverson, Stubblebine (1999)
hacs.nrl.navy.mil/publications/CHACS/1999/./1999syverson-fm99.ps

www.cs.bu.edu/techreports/pdf/2002-003-deanonymizing-safeweb.pdf
Deanonymizing Users of the SafeWeb Anonymizing Service - Martin, Schulman (2002)

http://citeseer.nj.nec.com/schneider95security.html
Security properties and CSP - Schneider (1995)

http://citeseer.nj.nec.com/bistarelli02relating.html
Relating Process Algebras and Multiset Rewriting (for Example for Security Protocol Analysis) (2002)

http://www.jya.com/onion.htm
JYA : Information about Onion Routing

http://citeseer.nj.nec.com/goldschlag99onion.html
Onion Routing for Anonymous and Private Internet Connections - Goldschlag, Reed, Syverson (1999)

http://citeseer.nj.nec.com/beimel01informationtheoretic.html
Information-Theoretic Private Information Retrieval: A Unified Construction

http://www.hipaadvisory.com/news/NewsArchives/2001/1214ncvhs.htm
Testimony for the National Committee on Vital and Health Statistics Standards and Security Subcommittee HIPAA-Readiness Survey Findings

http://www.hipaadvisory.com/views/archives/Provider/ahacomment21700.htm
AHA Comments on Privacy Regulation Proposal

http://www.hipaadvisory.com/news/Health%20Information%20Privacy%20On%20the%20Internet.htm
Health Information Privacy On the Internet

k-Anonymous Message Transmission Luis von Ahn,Andrew Bortz and Nicholas Hopper, Carnegie Mellon University